

FIG. 1

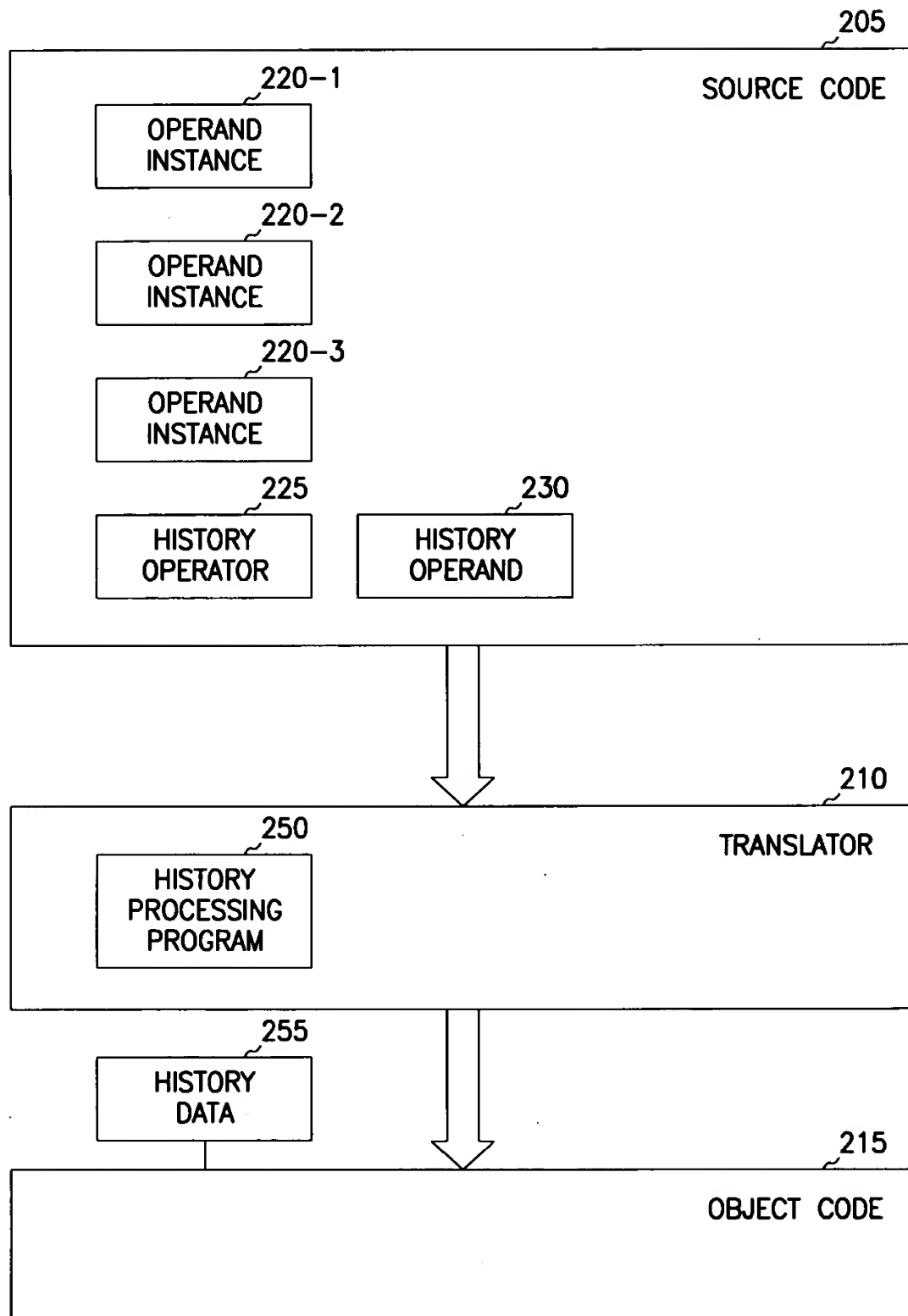


FIG. 2

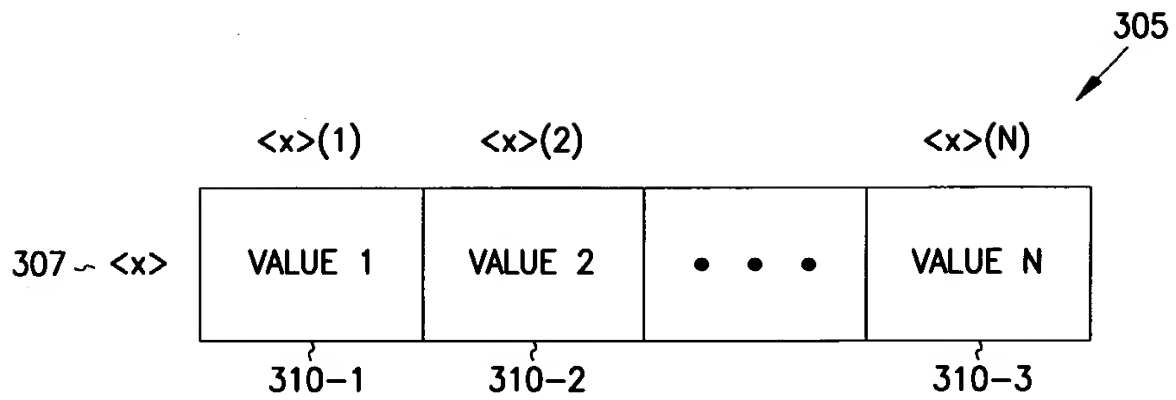


FIG. 3A

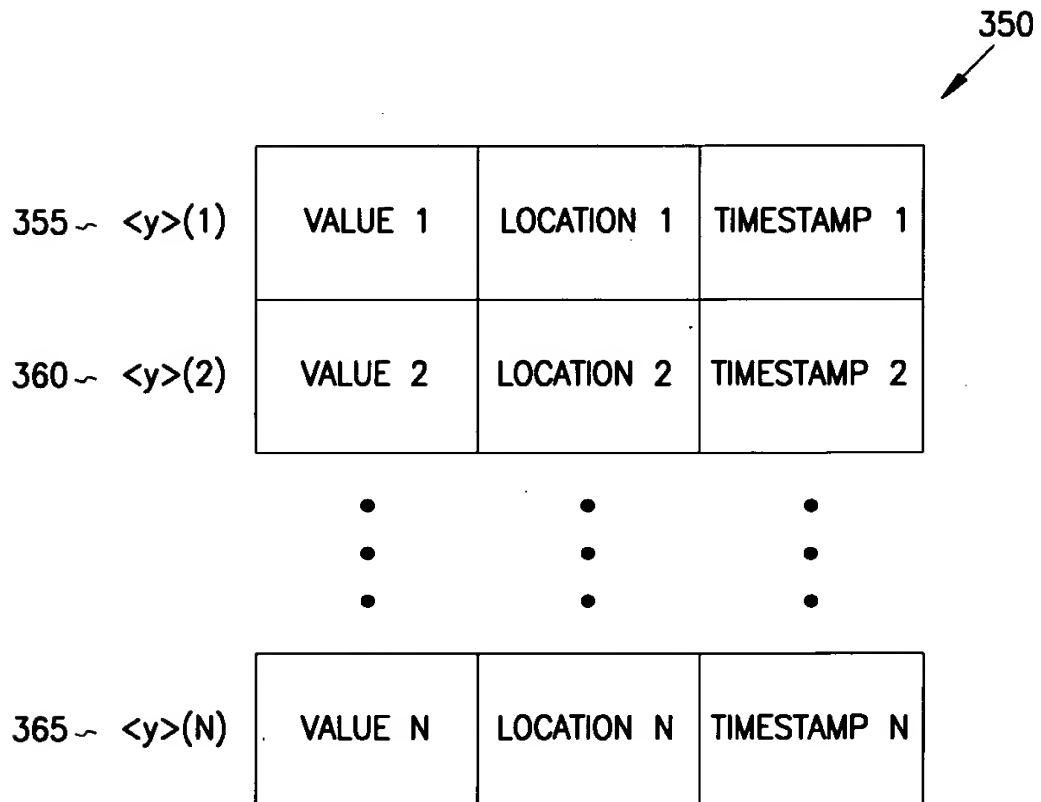


FIG. 3B

400

450

```
p = aList;
while (p != NULL) {
465 ~ x = p->value;
    p = p->tail;
}
print ("average %f\n", average<x>);
455 460
```

405

```
p = aList;
sum = 0;
count = 0;
while (p != NULL) {
    count += 1;
    sum += p->value;
    p = p->tail;
}
print ("average %f\n", sum/count);
```

FIG. 4

```

505
550
p = aList;
firstTime = true;
while (p != NULL) {
    if (firstTime) {
        firstTime = false;
    } else {
        printf (" ");
    }
    printf("%d", p->value);
    p = p->tail;
}

p = aList;
565 ~ while (p != NULL) { 560
    if (count < while > != 1) {
555 ~ printf(", ");
    }
    printf("%d", p->value);
    p = p->tail;
}

```

600

650

```
for (i = 0; i < ARRAY_SIZE; i++) {  
665 ~ x = a[i];  
        }  
        printf ("Max is %f\n", max<x>);  
660 {  
655 }
```

605

```
607 ~ int max = a[0];  
for (i = 0; i < ARRAY_SIZE; i++) {  
    if (max < a[i]) {  
        max = a[i];  
    }  
}  
printf ("Max is %f\n", max);
```

FIG. 6

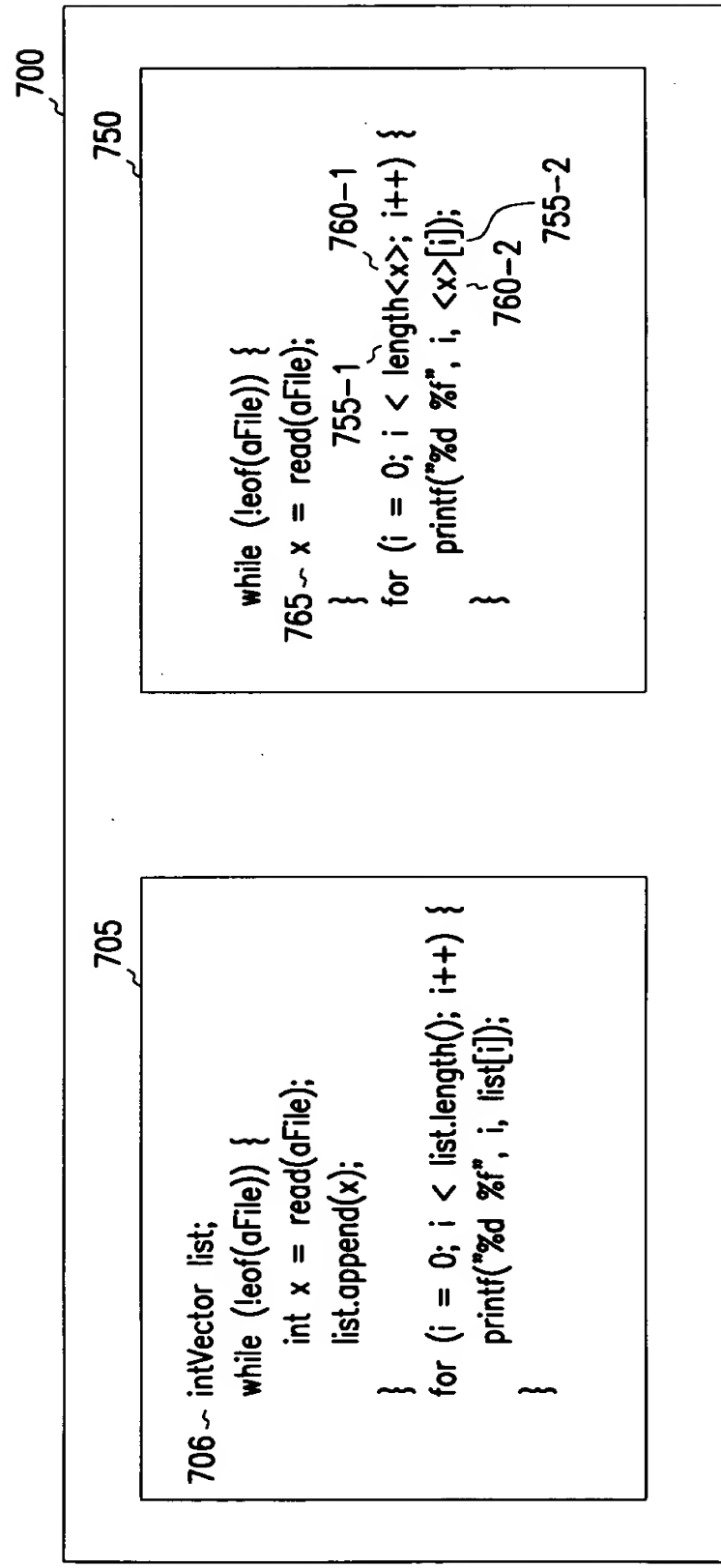


FIG. 7

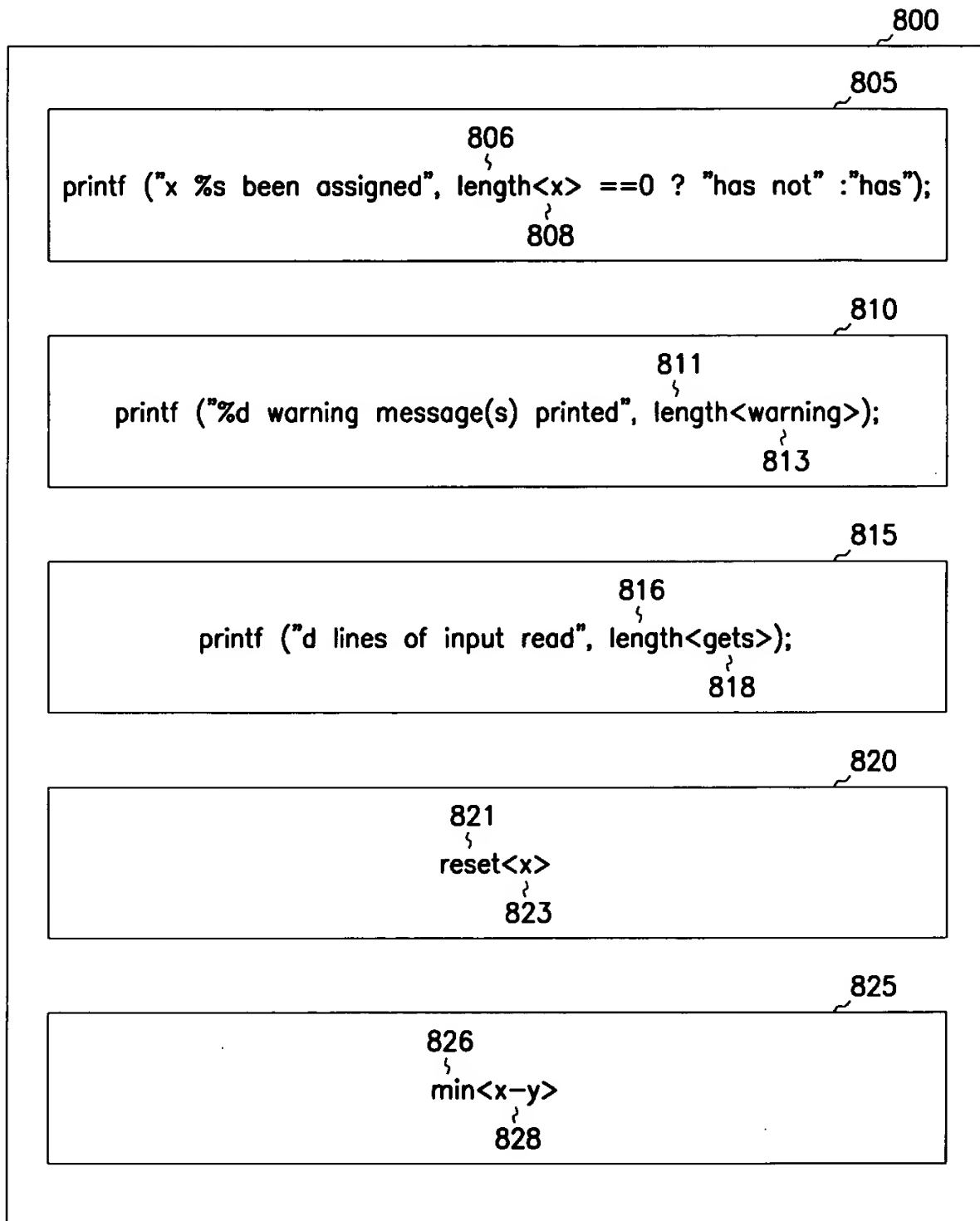
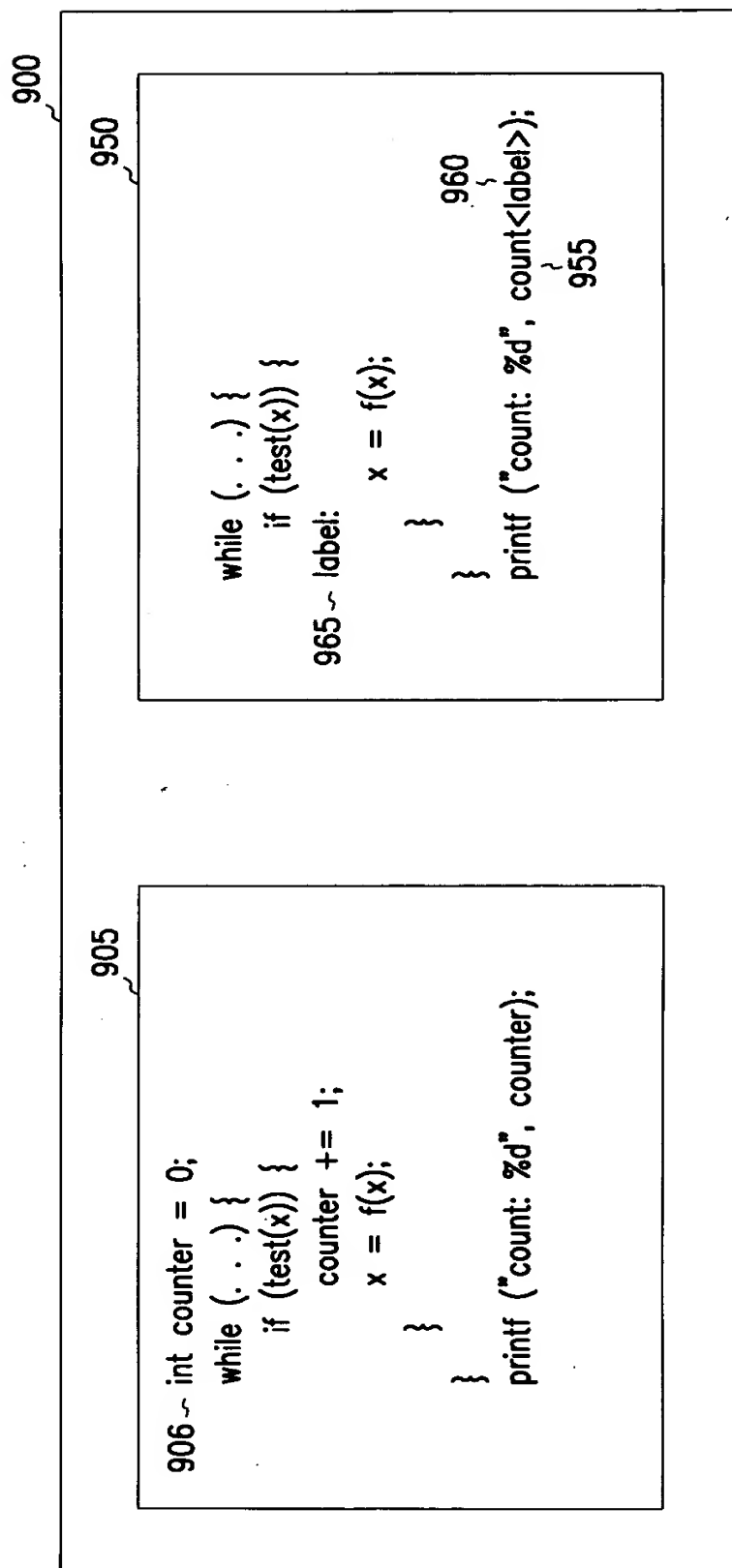
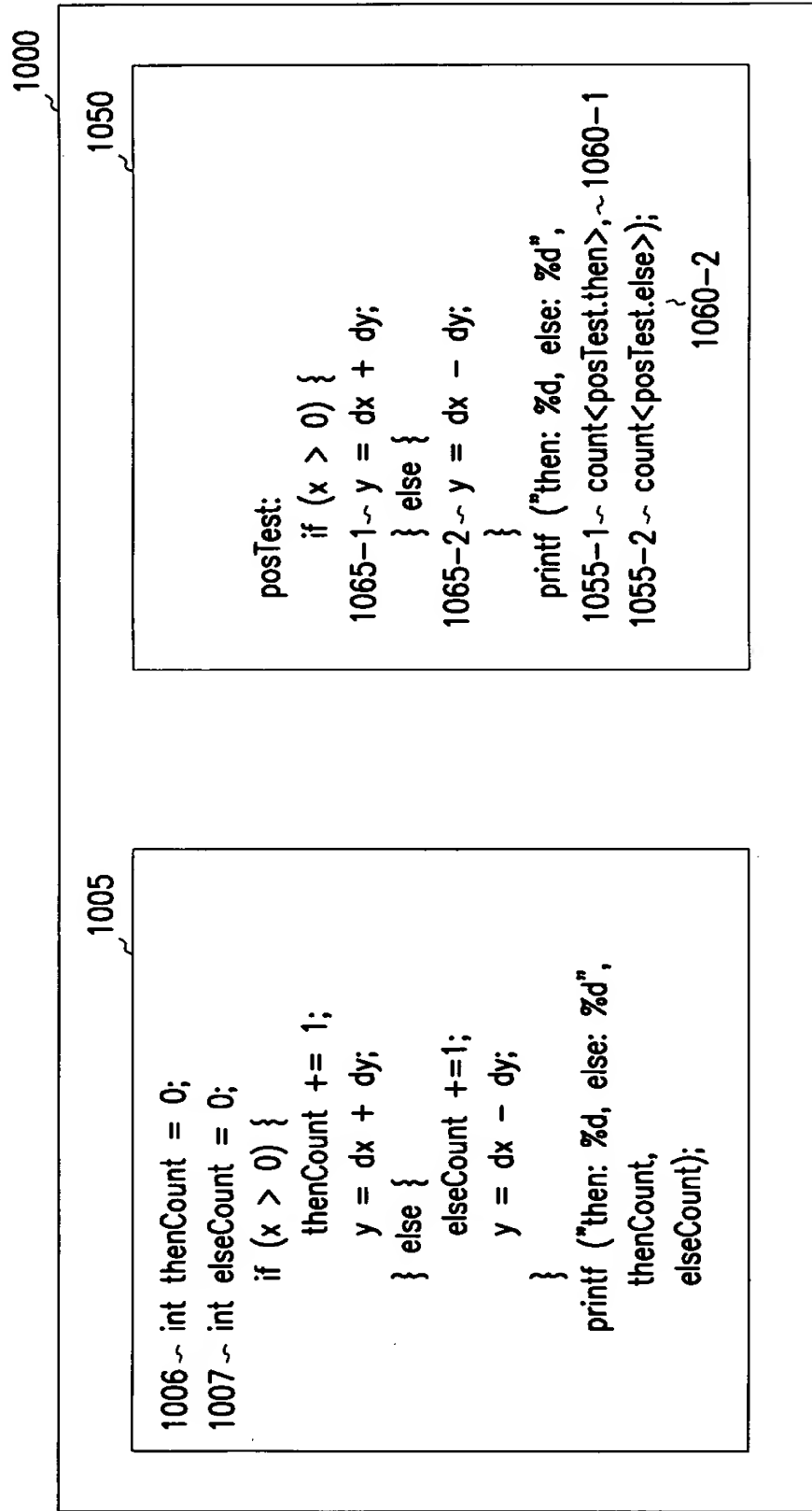


FIG. 8







**FIG. 10**

1100

1150

```

x = f(0);
do {
  x = f(x);
  if (count < while > > 10000) break;
} while (abs(x - prev<x>) > epsilon);
  
```

1105

```

1106 ~ int limit = 0;
x = f(0);
do {
  limit += 1;
  x = f(x);
  if (limit > 10000) break;
} while (abs(x - prev<x>) > epsilon);
  
```

FIG. 11

1200

```

p = aList;
while (p != NULL) {
    x = p.head();
    match:
        found = equal(p.head, key);
        if (found) break;
        p = p.tail();
    }
    print (searching required %d probes\n", length<match:equal>);

```

FIG. 12

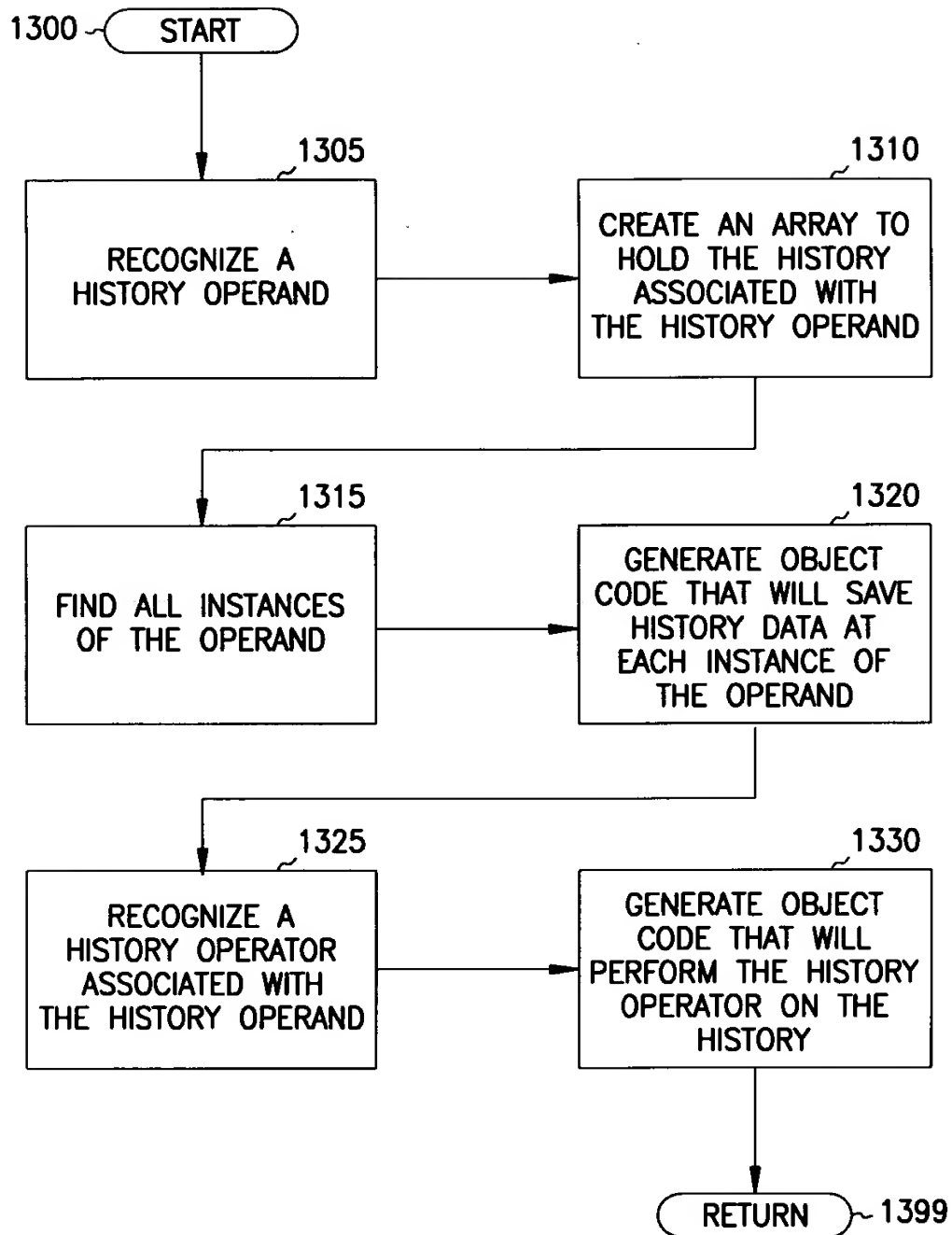


FIG. 13